Journal of Modern Technology and Engineering
Vol.7, No.3, 2022, pp.199-215

Jomard Publishing

A NEW APPROACH FOR SOLVING THE EUCLIDEAN STEINER TREE PROBLEM BASED ON DATA CLUSTERING

Duygu S. Turan* , **Burak Ordin**

Ege University, Faculty of Science, Department of Mathematics, Bornova, Izmir, Turkiye

Abstract. Steiner tree problem is a combinatorial optimization problem, which has many important applications in different areas such as medicine, communication, engineering. Several different algorithms have been developed for the solving of the problem that is NP-Complete. In this paper, new solution algorithms based on data clustering problem are proposed for approximate solution of Steiner tree problem. One of the main purposes of this study is to solve the combinatorial optimization problem with approaches in the field of data mining. The proposed methods are applied on 28 data sets in the literature and the results of preliminary numerical experiments are reported. Steiner tree problem is NP-Complete so that optimal solutions for some of these datasets, especially large scale data sets, can not obtain and are unknown. Therefore, importance of improving algorithms to find approximate solutions for such problems gradually increases.

Keywords: Text classification, TTC-3600 dataset, Machine learning, Methods with term weighting.
***Corresponding author:** Duygu S. Turan, Ege University, Faculty of Science, Department of Mathematics, Bornova, Izmir, Turkiye, e-mail: *duygu.selin.balli@ege.edu.tr*
Received: 12 October 2022; Revised: 17 November 2022; Accepted: 28 November 2022; Published: 29 December 2022.

1 Introduction

Steiner tree problem (STP) is based on finding shortest tree linking a set of points called terminal points in the plane. If a shorter tree is to be found when this shortest tree found, additional points called Steiner points can be used, except for the given terminal points (Chlebík & Chlebíková, 2008; Fampa & Anstreicher, 2008).

STP is an important graph and optimization problem which has many application areas. The problem has various applications from biology to communication, from electricity to several network design and has attracted considerable interest in the literature in recent years.

The Euclidean Steiner tree problem (ESTP) is a variation of STP. The distance between any two points is calculated using the Euclidean distance in Euclidean plane for ESTP. The ESTP is the shortest length tree finding problem that can be use additional points (Steiner points) if necessary, covering a set of points in Rd. The ESTP is also known as Steiner minimal tree. Additionally, the ESTP is an NP-hard problem. Therefore, the heuristic and approximate algorithms developed for the problem are of great importance Fampa & Anstreicher (2008). In Fampa et al. (2016), it is given an overview of the exact algorithms presented in the literature for the ESTP when $n \geq 3$ and discuss their common and distinguished features, their advantages and drawbacks, and some possible directions for improvement toward the numerical solution of large instances of the problem (Fampa et al., 2016).

First algorithm as a solution to the STP is presented in (Melzak, 1961). The Melzak algorithm is an algorithm that includes a finite number of steps that attempt to test a large number of possibilities. However, the algorithm works for very small number of terminal points.

Rectilinear distance was used in the article that published by Hanan in 1966 for STP and for the first time the rectilinear Steiner tree problem was mentioned (Hanan, 1966). In Gilbert & Pollak (1968) found conditions simplifying the task of finding the Steiner minimal tree for a large number of terminal points where the Melzak algorithm did not work well. Branch and bound algorithm was designed to find Steiner minimal tree in (Smith, 1992). Beasley presented a heuristic algorithm for Euclidean and rectilinear Steiner tree problem in 1992. The presented heuristic algorithm is based on finding optimal Steiner solutions for connected subgraphs of the minimal overlapping tree of all vertices. According to the results of problems with up to 10000 points and randomly generated, proposed heuristic gave better results compared to other heuristics (Beasley, 1992). In the article published by Karpinski and Zelikovsky in 1997, a new technique on selecting Steiner points depending on possible deviations from optimal for STP was proposed. With the proposed algorithm, performance ratios were obtained as 1.644 in the random metric and 1.267 in the linear plane (Karpinski & Zelikovsky, 1997). Koch and Martin presented an application of branch and cut algorithm for network Steiner tree problem and their proposed algorithm based on integer programming for oriented graph. Almost all problems from SteinLib library have been solved optimally (Koch & Martin, 1998). Qu and his colleagues presented first application of STP for multiple routing problem in 2013. The OR Library is used for the problem and the algorithm is based on the Jumping Particle Swarm Optimization in the literature. According to the experimental results, their proposed algorithm showed faster and better performance than many existing algorithms Xu et al. (2013). In Jones & Harris (1996) Steiner minimum spanning tree problem has been solved by a genetic algorithm. In Fampa & Anstreicher (2008) improvements of the branch and bound algorithm that proposed for ESTP by Smith in 1992 is presented.

There are many variations of the STP. Various solution algorithms have been developed for each variation. Although there are algorithms that provide the optimal solution for a few terminal points, the importance of developing approximate and heuristic algorithms for problems involving a large number of terminal points is increasing (Hakimi, 1971; Winter & Zachariasen, 1996).

The data clustering problem, which is another combinatorial optimization problem, deals with the problems of organization of a collection of patterns into clusters based on similarity. It is among the most important tasks in data mining. K-means algorithm that is one of the various heuristics has been developed to tackle clustering problem. The method is simple and fast. Besides, an other method, the global k-means algorithm, introduced in Likas et al. (2003) is a significant improvement of the k-means algorithm. In this algorithm each data point is used as a starting point for the k-th cluster center. Such an approach leads to a global or near global minimizer (Ordin & Bagirov, 2015).

In this paper, the solution methods of the data clustering problem in data mining is used for the ESTP, which is a kind of STP. Methods based on the clustering are proposed for the solving of the ESTP with given n terminal points. The algorithms which are used in this methods are k-means and incremental version of the k-means algorithm. The aim of the proposed methods are to find shortest tree connecting n terminal points with k Steiner points.

The rest part of the paper is organized as follows: In Section 2, STP and solution algorithms of the STP are introduced. In Section 3, the data clustering problem is formulated and implementation of the proposed algorithms are explained. In Section 4, the solution of the ESTP is designed by clustering technique. The results of numerical experiments are given in Section 5 and Section 6 concludes the paper.

2 Steiner Tree Problem and its solution

The first version of the STP which proposed by Pierre de Fermat is from the 1600s. It is related to finding of a point with a minimum distance of given three points in a plane. In other words,

for a given T triangle with vertices a_1, a_2, a_3, there is such a point in the T field that minimize the sum of distances $|pa_1|+|pa_2|+|pa_3|$. In this case, the point is called the Steiner point and is unique. Between 1608-1647, Evangelista Torricelli found a geometric solution for this problem (Melzak, 1961).

The problem, which is a generalization of the Fermat problem, can be expressed as:

Given a P convex polygon with vertices $a_1, a_2, ..., a_n$. Find a q point in the P field so that you can minimize the $\sum_{i=1}^{n} |qa_i|$ sum (Melzak, 1961).

The general Fermat problem, known as the search for a point with a minimum distance to the n points in the plane, has attracted the attention of a group of famous mathematicians, including Jacob Steiner. This problem is generalized to the following:

Let the points $a_1, a_2, ..., a_n$ be $n \geq 3$ in a plane. A tree is found so that, it is the shortest tree that contains n points (Melzak, 1961). The shortest length obtained is named Steiner minimal tree. In this tree, $a_1, a_2, ..., a_n$ are called terminal points. Steiner minimal tree, in addition to $a_1, a_2, ..., a_n$, may contain other points and these points are called Steiner points(Gilbert & Pollak, 1968).

The STP has different variants for different metrics. These various problems are derived from different types of applications. Let $u = (u_x, u_y)$ and $v = (v_x, v_y)$ be two points. The distance between points u and v in the L_p metric is $|uv|_p = (|u_x - v_x|^p + |u_y - v_y|^p)^{(1/p)}$, where $1 \leq p < \infty$. For example, L_2 metric is $|uv|_2 = \sqrt{|u_x - v_x|^2 + |u_y - v_y|^2}$ and this metric is called as Euclidean distance. STP where this norm is used are called ESTP (Gilbert & Pollak, 1968).

The STP has three basic types: the ESTP, the rectilinear STP and the network STP

The network STP, known as the STP in graphs, involves combining a subset of the set of vertices with a minimum cost. More specifically, given an undirected connected graph G=(V,E) with vertex set V, edge set E, nonnegative weights associated with the edges, and a subset B of V (a set of terminal points). The problem is to find a subgraph T that connects the vertices in B so that the sum of weights of edged in T is minimised. Obviously, the solution is a tree and this tree is called as Steiner minimum tree for B in G. If $|B| = 2$ (there are 2 terminal points), then the problem is reduced to shortest path problem and can be solved with the Dijkstra algorithm. If $B = V$ (there is no any Steiner points), then the problem is reduced to a minimum spanning tree problem and can be solved with the Prim, Boruvka or Kruskal algorithm. These two problems are polynomial, but the network Steiner tree problem is an NP-complete problem, and therefore developing heuristic and approximate methods for large data sets is important. Let $V = 1, 2, \ldots, n$ and S be a set of Steiner points. For each (i, j) edge, $c_ij \geq 0$ is a weight of the edge. The aim of network STP is to find a connected graph $G^{'} = (B \cup S, E^{'})$ (Steiner tree) with $E^{'} \subset E$, for the sum of weights to be minimal. A bivalent variable x_ij for each edge $(i, j) \in$ is defined: if the edge (i, j) is included into the Steiner tree, then $x_ij = 1$ if not $x_ij = 0$. Similary another bivalent variable f_i is defined: if vertex i is included in the Steiner tree, then $f_i = 1$ if not $f_i = 0$. y_ij represents the flow through the edge $(i, j) \in E^{'}$, mathematical formulation of the network STP is derived as follows (Seda, 2001):

$$Minimize \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

subject to

$$r := mink|k \in B \tag{2}$$

$$\forall j \in (V - r) : x_{rj} = 0 \tag{3}$$

$$\forall i \in (V - r) : \sum_{j=1}^{n} x_{ij} = f_i \tag{4}$$

$$\forall i \in (V - r) : \sum_{j=1}^{n} y_{ij} - \sum_{j=1, j \neq r}^{n} = f_i \tag{5}$$

$$\forall i, j \in V : y_{ij} \leq (n-1)x_{ij} \tag{6}$$

$$\forall i, j \in V : y_{ij} \in Z_+ \tag{7}$$

$$\forall i, j \in V, c_{ij} = 0 : x_{ij} := 0 \tag{8}$$

$$\forall i, j \in V : x_{ij} \in 0, 1 \tag{9}$$

$$\forall i \in B : f_i = 1 \tag{10}$$

$$\forall i \in (V - B) : f_i \in 0, 1 \tag{11}$$

where Z_+ denotes the set of nonnegative integers.

As mentioned earlier, the network STP is an NP-complete problem, so approximate and heuristic approaches are needed for large-scale examples. The quality of an approximate algorithm is measured by the ratio of the obtained solution to the optimum solution, that is, the performance ratio. The most known approximate algorithms for the network Steiner tree problem are the distance network approximation, the minimum path appriximation and the contraction approximation (Seda, 2001).

There are many versions of the STP , and these versions are usually named according to the distance metric. In rectilinear Steiner tree problem, the distance $d(v_i, v_j)$ between two v_i and v_j points is defined by rectilinear (Manhattan) metric (Seda, 2001):

$$d(v_i, v_j) = |x_i - x_j| + |y_i - y_j| \tag{12}$$

where (x_i, y_i) are the Cartesian coordinates of v_i.

In all of these, the rectilinear Steiner tree is the shortest network of horizontal and vertical lines connecting all the terminals of V. This problem is also an NP-complete problem so it is very important to have an approximate algorithm that can reasonably solve this problem.

The ESTP asks for the shortest planar straight-line spanning the set $V = v_1, v_2, \ldots, v_n$ in the Euclidean plane. The elements of the set V are called as terminals. The solution has form of a tree. Unlike the minimum spanning tree problem, in the ESTP the connections do not have to be just between the terminal points. A shorter network can be obtained with additional points called Steiner points. In the ESTP, the distance $d(v_i, v_j)$ between two v_i and v_j points is calculated as follows (Seda, 2001):

$$d(v_i, v_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{13}$$

The first mathematician to design an algorithm for the ESTP is Melzak Soothill (2010). The algorithm presented by Melzak is an exact algorithm but it is not applicable for large n values because it involves testing a large number of probabilities (Melzak, 1961). Smith's numerical algorithm is similar to the Melzak algorithm in Euclidean space. The advantage of this algorithm is that it can be easily generalized to higher dimenstions as opposed to the Melzak algorithm. Up to 18 years ago, the problems that exact algorithms can solve include up to 29 terminal points. Today, the GeoSteiner algorithm is the best exact algorithm that can solve the ESTP. This algorithm can solve problems with up to 2000 terminal points (Soothill, 2010).

The STP concerns the finding of a minimum length tree connecting n terminal points using additional points if necessary. For $n = 3$, the problem becomes Fermat problem. The problem is becoming exponential due to the increasing value of n. There is no known algorithm that solves this problem in polynomial time. Heuristic algortihms are built using minimum spanning trees. Looking at all these, the importance of improving approximate algorithms for the STP is increasing (Dreyer & Overton, 1998).

The STP is one of the most important graph problems because it has many applications area. One application of the Steiner tree is that of the minimal network theory. The most popular problem in the minimal network theory is the finding of absolute minimal Networks

spanning a set of points. Another application, Multicasting Routing, aims to effectively connect a set of destinations in a network for intra-group communications such as teleconferencing. In addition to these, there are areas such as modeling the evolution of species in biology, modeling soap films for cable grids, creation of oil and gas pipelines, communication Networks, electricity distribution networks, design of roads and train tracks.

3 Data clustering problem and its solution

Data Clustering is one of the most important methods of data mining, and is used in many areas to provide meaningful and useful information. In other words, a data set is divided into meaningful subsets in clustering. The need to produce meaningful and useful output is not only for data mining, but also for many areas such as pattern recognition, statistics, medicine, and marketing. For this reason, clustering analysis is an interdisciplinary field of study with applications in various fields.

In the clustering technique there is no label information about the cluster of classes, contrary to the classification technique, because clustering is a problem of unsupervised learning. In the clustering analysis, the process of assigning the data to the subclusters in a meaningful manner is based on the similarity or distance measure between data. Correspondingly, it is expected that the similarity between data in the same cluster as a result of clustering is greater than the similarity between data in different clusters (Hartigan & Wong,, 1979; Xu & Wunsch, 2005).

The mathematical model of the clustering problem is as follows:

Let A be a set of n dimensional m points in the space R^n such that $A = a^1, \ldots, a^m$, $i = 1, \ldots, m$ and $a^i \in R^n$. The purpose of the clustering problem is to divide the data in the set A into k clusters A^j for $j = 1, .., k$ satisfying the following conditions (Bagirov & Mardaneh, 2006):
1) $A^j \neq \emptyset$, $j = 1, ..., k$;
2) $A^j \cap A^l = \emptyset, l = 1, ..., k, j \neq l$;
3) $A = \cup_{j=1}^{k} A^j$.
4) There is no constraint on A^j clusters for $j = 1, ..., k$. $A^j, j = 1, ..., k$ are called as clusters. Each A^j cluster can be represented $x^j \in R^n$ centers for $j = 1, ..., k$.
The distance between data points is calculated by a measure called "similarity measure". This measure is defined by the distance to the center of a data point. A clustering problem can be reduced to an optimization problem as follows:
Let $d(x, y)$ be the distance between the x and y points.

$$Min : \psi_k(x, w) = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{k} w_{ij} d(x^j, a^j) \tag{14}$$

subject to

$$x = (x^1, ..., x^k) \in R^{n \times k}, \tag{15}$$

$$\sum_{j=1}^{k} w_{ij} = 1, i = 1, ..., m \tag{16}$$

and

$$w_{ij} = 0 or 1, i = 1, ..., m, j = 1, ..., k \tag{17}$$

where w_{ij} is the association weight of pattern a^i with cluster j, given by

$$w_{ij} = \begin{cases} 1, & \text{if pattern } a^i \text{ is assocated to cluster } j \\ 0, & \text{otherwise} \end{cases}$$

and

$$x^j = \frac{\sum_{i=1}^{m} w_{ij} a^i}{\sum_{i=1}^{m} w_{ij}}, j = 1, ..., k.$$

Here w is an $m \times k$ matrix. Nonsmooth nonconvex optimization formulation of the clustering problem is as follows (Ordin & Bagirov, 2015):

$$Min : f_k(x) = \frac{1}{m} \sum_{i=1}^{m} min_{j=1,...,k} d(x^j, a^i) \tag{18}$$

subject to

$$x = (x^1, ..., x^k) \in R^{n \times k}, \tag{19}$$

ψ_k and f_k are called as cluster functions.

Many algorithms have been developed for solving the clustering problem. Clustering algorithms can be divided into five classes in general: partitioned methods, hierarchical methods, density based methods, grid based methods and model based methods. In the partitioned clustering method, which is the clustering technique used in this article, n data points are partitioned into k clusters for $k \leq n$. As a result of this operation, each data point belongs to only one cluster. In this clustering method, only one result is produced depending on the value k. It is the most known k-means algorithm of partitioned clustering methods. We are given a training set $x^1, \ldots, x^n$ and the number of cluster k. The pseudo-code of the k-means clustering algorithm is the following (MacQueen, 1967)

Algorithm: PROGRAM K-MEANS

1. INITIALIZE cluster centers $c_i, ..., c_k$
2. REPEAT
3. Set $i = 1$;
4. WHILE $i \leq n$
5. Assign the data i to its closest cluster using

$$z_{ij} = \begin{cases} 1, & \text{if } j = argmin_j ||x^i - c_k||^2 \\ 0, & \text{otherwise} \end{cases}$$

6. Set $i = i + 1$
7. END WHILE
8. Set $j = 1$;
9. WHILE $j \leq k$
10. Compute the number of data assigned cluster j using $n_j = \sum_{i=1}^{n} z|ij$;
11. Update the cluster j using $c_j = \frac{1}{n_j} \sum_{i=1}^{n} z_{ij} x^i$;
12. Set $j = j + 1$;
13. END WHILE
14. UNTIL Convergence
15. RETURN final solution $c_1, ..., c_k$;
16. END PROGRAM K-MEANS

In the k-means algorithm, after the number of clusters is determined as k, the initial cluster centers is selected to represent this k clusters. Then the distances of each data point to these k clusters are calculated by the selected similarity-distance function and the closest cluster assignments are made. After the assignment is finished, new cluster centers are calculated for each cluster. As long as the stopping criterion is not satisfied, the distance between the data points and the centers is calculated and reassignment process continues. Except for the clustering algorithms mentioned above, there are several variants of the clustering algorithms. We are given a training set $x^1, \ldots, x^n$ and the number of cluster k. The pseudo code of the incremental k-means algorithm which is a variation of the k-means algorithm, is as follows (Likas et al., 2003)
Algorithm: PROGRAM GLOBAL K-MEANS

1. INITIALIZE first cluster using $m_1 = \frac{1}{n}\sum_{i=1}^{n} x^i$;
2. Set $t = 1$;
3. WHILE the termination condition $t > k$ is not met
4. Set $p = 1$;
5. WHILE $p \leq n$
6. Run the k-means algorithm for initial cluster centers $(m_1, ..., m_{t-1}, x^p)$
7. IF the obtained clustering results is better than the previous one THEN
8. Set $y_i = m_i$ for $1 \leq i \leq t-1$ and $y_t = m_t$;
9. ENDIF
10. Set $p = p + 1$;
11. END WHILE
12. Set $m_i = y_i$ for $1 \leq i \leq t$ and $t = t + 1$;
13. END WHILE
14. RETURN final solution $m_1, ..., m_k$;
15. END PROGRAM GLOBAL K-MEANS

The incremental k-means algorithm aims to find the best solution at every iteration.In the first step of the algorithm, the mean of all data in the data set is calculated, the first cluster center is determined and $k = 1$ assignment is made. In step 2, it is checked whether the number of elements in the data set has been reached, and if so, the algorithm is stopped. If not, all elements of the dataset are tried one by one for the k^{th} cluster center. The k-means algorithm given above is applied and the best one is recorded.Internal quality measurement methods can be used to decide which is best.Incremental clustering algorithms generally produce results with low accuracy in small data sets. On the other hand, they give better and more effective results on large data sets.

4 On the solutions of ESTP based on data clustering algorithms

The STP deals with finding the shortest-length tree that connects a given set of terminal points using additional Steiner points if necessary. There are various variations of the STP and are usually named according to the used metrics. As mentioned in the Section 2, the STP using Euclidean distance (L2-distance) is called the ESTP and this problem is NP-complete problem.Therefore, the problem becomes a structure that can not be solved as the number of terminal points increases.For this reason, it is very important to develop heuristic or approximate algorithms for the problem.

In this article, two approximate solution method is designed for the ESTP. Additional Steiner points can be used to solve this problem except for the terminal points. In the proposed methods, the steiner points to be used are determined by the clustering technique. The k-Means algorithm which is most frequently used as clustering algorithm and the incremental k-Means algorithm are used. When a data set with n-terminal points is clustered by the k-means clustering algorithm, terminal points are separated into several clusters. Clustering results are obtained in which the number of clusters changes from 1 to n and the result with the shortest distance is recorded. In the incremental k-means clustering algorithm, since the number of clusters is not determined from the beginning, clustering from 1 to n is already done and the best is recorded. The cluster centers obtained as a result of clustering are called Steiner points in the ESTP.

In order to get Steiner tree, firstly we connect the n points to the center each of the points belongs. Thus we obtain k star graphs. Some basic definitions and theorems about graph are given below (Guichard, 2017).

Definition 1. *A connected graph G is a tree if it has no cycles.*

Definition 2. *A star graph $K_{1,n}$ is a tree containing exactly o vertex is not pendent vertex.*

Theorem 1. *A tree with n vertices has exactly $n-1$ edges.*

Current graph is not connected so it cannot be a tree. We use Prim's algorithm to connect these star graphs. For this connection process, non-pendent vertices of each star graphs are utilized and these vertices are connected using Prim's algorithm. Since on Euclidean space, the distances between these vertices are calculated by the Euclidean distance.

Theorem 2. *The Prim's algorithm produces a minimum cost spanning tree Guichard (2017).*

With the help of the definitions and theorems stated above, the following theorem can be put forward.

Theorem 3. *Let K^j_{1,n_i} be star graphs, for $i, j = 1, \ldots, k$. If we connect the non-pendent vertices with Prim's algorithm, then we get a tree.*

Proof. From Definition 1, K^j_{1,n_i} is a tree, for each $i, j = 1, \ldots, k$. Therefore, it is connected and contains no cycle with Definition 2. The degree array of vertices of K^j_{1,n_i} is $(n_i, 1, \ldots, 1)$. In other words, the tree has one vertex with degree n_i and n_i vertices with degree 1.
Let's prove that if we connect the vertices with degree n_i of each tree using Prim's algorithm, we get a tree. Let the name of this tree be $T(V, E)$. The tree T has n vertices and $n = n_1 + n_2 + ... + n_k$.
$K^j_{1,n_i}(V_i, E_i)$ is a tree, so $|V_i| = n_i + 1$ and $|E_i| = n_1$ with Theorem 1. If we connect the k vertices with Prim's algorithm, we have to add $k-1$ edges. In that case, T have $n_1 + ... + n_k + k$ vertices and $n_1 + ... + n_k + k - 1$ edges. It is clearly seen that $|V| = |E| + 1$ for the tree T and so T is a tree. □

In order to calculate the distance of the obtained tree, firstly, the total distance of the n points to the center where each of the points belongs is calculated by the Euclidean distance. Then the k cluster centers obtained by the clustering are connected to each other using Prim's algorithm. The pseudo-code of the Prim algorithm for a graph $G = (V, E)$ is the following (Bondy & Murty, 1976):
Algorithm: PROGRAM PRIM

1. Select an arbitrary vertex $v \in V$
2. Set $key_u = \infty$ for each $u \in V$ and $Q = V$;
3. Set $key_v = 0$ and $p_v = NULL$;
4. WHILE $Q = \emptyset$ is not met
5. Find a vertex $s \in Q$ incident on a shortest edge connecting with a vertex in the tree
6. Add s into the tree and remove s from Q
7. FOR EACH vertex w adjacent to s DO
8. IF $w \in Q$ and key_s is greater than a_{sw} the weight of edge (s, w) THEN
9. Set $key_s = a_{sw}$;
10. Set $p_w = s$;
11. ENDIF
12. END FOR EACH
13. END WHILE
14. RETURN final solution;
15. END PROGRAM PRIM

After k cluster centers, ie. Steiner points, are connected by Prim Algorithm, we obtain the Steiner tree and the total length of the tree is found.

We call the method in which the k-means algorithm is used as the KESTP. In the direction mentioned above, the pseudo-code of the KESTP for n terminal points is the following:
Algorithm: PROGRAM KESTP

1. INITIALIZE set $k = 1$;
2. WHILE $k > n$ is not met
3. Run the k-means algorithm for n terminal points and k
4. Find the total distance of each terminal points to the cluster to which it belongs and assign this value to the $distance1$ variable
5. Find the minimum distance tree spanning the k cluster centers with Prim's algorithm, and assign the obtained distance to the $distance2$ variable.
6. Set $distance = distance1 + distance2$;
7. IF $(k = 1)$ or $(shortest > distance)$ THEN

8. Set $shortest = distance$;
9. Keep the centers as Steiner points
10. ENDIF
11. Set $k = k + 1$;
12. END WHILE
13. RETURN final solution *shortest* and Steiner points;
14. END PROGRAM KESTP

Similarly, we use the method to solve the ESTP with incremental k-means clustering algorithm and call this method as the GKESTP. The pseudo-code of the GKESTP for n terminal points is the following:

Algorithm: PROGRAM GKESTP

1. INITIALIZE set $k = 1$;
2. WHILE $k > n$ is not met
3. Run the incremental k-means algorithm for n terminal points and k
4. Find the total distance of each terminal points to the cluster to which it belongs and assign this value to the $distance1$ variable
5. Find the minimum distance tree spanning the k cluster centers with Prim's algorithm, and assign the obtained distance to the $distance2$ variable.
6. Set $distance = distance1 + distance2$;
7. IF $(k = 1)$ or $(shortest > distance)$ THEN
8. Set $shortest = distance$;
9. Keep the centers as Steiner points
10. ENDIF
11. Set $k = k + 1$;
12. END WHILE
13. RETURN final solution *shortest* and Steiner points;
14. END PROGRAM GKESTP

The centers stored in KESTP and GKESTP are Steiner points for the Euclidean Steiner problem. Obviously, the only difference between KESTP and G KESTP is the clustering algorithm that applied. After finding the centers (Steiner points) using the appropriate clustering algorithm both in KESTP and in GKESTP, the distance from each terminal point to the cluster center that it assigned is calculated. Then, the centers are connected to each other by the Prim's algorithm. The total distance of the resulting tree (by using Theorem 3) is calculated, and the total distance is compared with the distances obtained from the previous iterations.

As a result of this comparison, the information of the tree with the shortest distance is stored. When the algortihm is terminated, you have the knowledge of the Steiner points which will give the approximate shortest distance.

The obtained distance is an approximate solution for the problems taken from the OR Library and the error are calculated to assess how well these solutions are found. Optimal solutions of the problems are in the library. The following formula is used to calculate this error:

$$error = \frac{\triangle_{clustering} - \triangle_{best}}{\triangle_{best}} \times 100$$

Here, the $\triangle_{best}$ value is the best known value available from the OR Library Winter & Zachariasen (1996) and the $\triangle_{clustering}$ value is the distance value obtained by the clustering algortihm used.

5 Experimenral Results

To verify the efficiency of the proposed algorithms numerical experiments with 28 real-world data sets have been carried out on a PC Intel(R) Core(TM) i3 with CPU 2.13 GHz and RAM 3 GB running under Windows 7. We used problems from OR Library to test our methods. We got 28 of problems with various number of terminal point. Selected data sets contain up to 1000 terminal points. In the OR Library,the estein1 file in the test problems for the ESTP contains 46 problems with different number of terminal points. There are also 10 different files, each with 15 problems, with 10, 20, 30, 40, 50, 60, 70, 80 and 100 terminal points respectively. For example, the file containing 40 terminal points has been named estein40. For the sake of diversity, 17 problems in the file estein1 and 5 problems from the other 10 files, two from each, have been tested. However, there are problems in the library that do not have optimal values. These problems have 250, 500, 1000 and 10000 terminal points.

In the experiments, 6 of the problems with 250, 500 and 1000 terminal points were tested, two from each, and the results were presented. For the 22 problems with optimal values, the proposed methods and their optimal values are compared and the errors are given in the tables. For the 6 problems with no optimal value, a comparison was made between the proposed methods (Beasley, 2017).

The results of experiments are presented in the following tables. In tables, the name "Esteinx (n)" in the problem name part refers to the n^{th} problem in the "esteinx.txt" file in the OR Library (Beasley, 2017).

In Figure 1, experiments were performed for 22 problems selected from the OR Library. These selected problems have the optimal values (best known values). Obtained results with KESTP for these 22 problems were compared with their optimal values. Comparisons were made with the error rate mentioned in Section 4. The number of Steiner points for optimal values and the number of Steiner points when KESTP was applied are also given in the table. In addition to these, the execution time for the KESTP algorithm is presented in the table. In experiments with KESTP, relation between the number of Steiner points and error is visualized in Figure 2.

In Figure 3, experiments were performed for 22 problems mentioned in Table 1. Obtained results with GKESTP and their optimal values were presented in the table. Comparisons between obtained results and optimal values were made with error rate mentioned in Section 4. The number of Steiner points for optimal values and the number of Steiner points when GKESTP was applied are also given in the table. In addition to these, the execution time for the GKESTP algorithm is presented in the table. In experiments with KESTP, relation between the number of Steiner points and error is visualized in Figure 4.

In Figure 5, experiments were performed for 6 problems selected from the OR Library. There are no optimal values of these 6 problems. We run the KESTP and GKESTP algorithms for

Table 1: Obtained results with the KESTP for known optimal value which of datasets

Problem Name	n **Terminal Point Number**	$\triangle_{clustering}$ **Optimal Value**	**Optimal Steiner Point Number**	$\triangle_{k-means}$ **Resulting Value by k-means Algorithm**	**Steiner Point Number for Proposed Method**	**Execution Time (sn)**	**Error (%)**
Estein1(20)	3	1.039615	1	1.039615	1	0.011	%0.00
Estein1(29)	3	1.465977	1	1.465977	1	0.014	%0.00
Estein1(22)	4	0.503286	1	0.512645	0	0.018	%1.86
Estein1(36)	4	0.878912	2	0.900000	0	0.010	%2.40
Estein1(1)	5	1.664399	2	1.721413	4	0.013	%3.42
Estein1(21)	5	1.818179	3	1.867506	0	0.015	%2.71
Estein1(5)	6	2.044052	3	2.044245	0	0.017	%0.00
Estein1(10)	6	1.598752	4	1.647427	0	0.018	%3.04
Estein1(3)	7	2.077671	0	2.077671	0	0.017	%0.00
Estein1(9)	7	1.559423	3	1.649300	4	0.020	%5.76
Estein1(4)	8	2.138789	0	2.138789	0	0.019	%0.00
Estein1(12)	9	1.648338	3	1.720820	8	0.023	%4.40
Estein1(40)	10	1.417988	6	1.485954	9	0.022	%4.79
Estein1(8)	12	2.177795	7	2.303096	10	0.031	%5.75
Estein1(31)	14	2.332174	6	2.512403	13	0.037	%7.73
Estein1(43)	16	2.330765	7	2.496152	11	0.044	%7.10
Estein1(33)	18	2.225805	8	2.398826	13	0.057	%7.77
Estein20(10)	20	3.011873	7	3.304484	14	0.054	%9.72
Estein40(5)	40	4.543251	13	5.390350	27	0.158	%18.65
Estein70(5)	70	5.476696	26	6.521155	47	0.416	%19.07
Estein90(10)	90	6.320064	36	7.851109	47	0.653	%24.22
Estein100(10)	100	6.719510	38	8.876331	56	0.925	%32.09

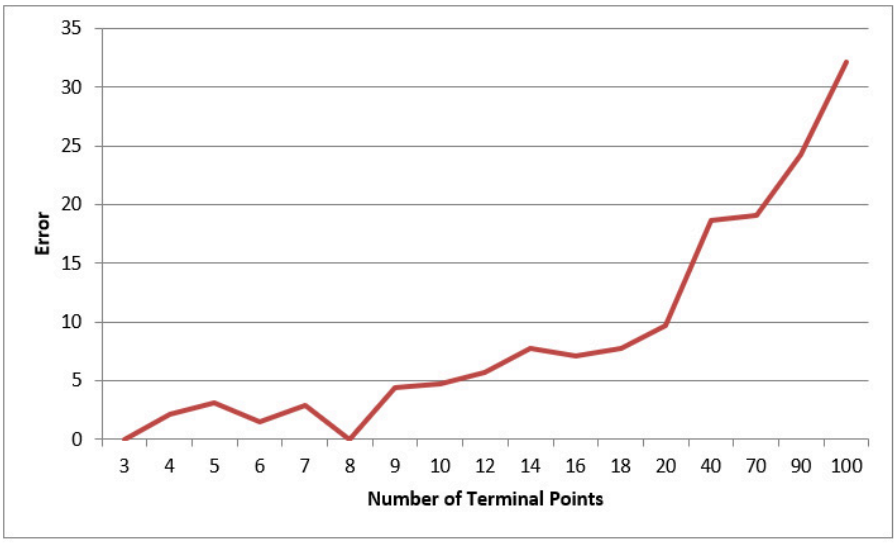

Figure 1: The relation between the number of the terminal points and error for the KESTP.

Table 2: Obtained results with the GKESTP for known optimal value which of datasets.

Problem Name	n **Terminal Point Number**	$\triangle_{clustering}$ **Optimal Value**	**Optimal Steiner Point Number**	$\triangle_{globalk-means}$ **Resulting Value by k-means Algorithm**	**Steiner Point Number for Proposed Method**	**Execution Time (sn)**	**Error (%)**
Estein1(20)	3	1.039615	1	1.039615	1	0.036	%0.00
Estein1(29)	3	1.465977	1	1.484924	2	0.058	%0.12
Estein1(22)	4	0.503286	1	0.539508	1	0.065	%7.19
Estein1(36)	4	0.878912	2	0.901141	3	0.028	%2.52
Estein1(1)	5	1.664399	2	1.794989	3	0.037	%3.42
Estein1(21)	5	1.818179	3	1.966274	2	0.040	%8.14
Estein1(5)	6	2.044052	3	2.264124	2	0.055	%10.76
Estein1(10)	6	1.598752	4	1.711342	5	0.082	%7.04
Estein1(3)	7	2.077671	0	2.519062	2	0.106	%21.24
Estein1(9)	7	1.559423	3	1.680044	4	0.094	%7.73
Estein1(4)	8	2.138789	0	2.627479	4	0.087	%22.84
Estein1(12)	9	1.648338	3	1.870030	4	0.146	%13.44
Estein1(40)	10	1.417988	6	1.489637	7	0.140	%5.06
Estein1(8)	12	2.177795	7	2.469291	7	0.192	%13.38
Estein1(31)	14	2.332174	6	3.018730	6	0.331	%29.43
Estein1(43)	16	2.330765	7	2.923562	5	0.371	%25.43
Estein1(33)	18	2.225805	8	2.858205	7	0.508	%28.41
Estein20(10)	20	3.011873	7	3.476368	10	0.655	%15.42
Estein40(5)	40	4.543251	13	5.413266	21	2.990	%19.14
Estein70(5)	70	5.476696	26	7.238848	26	17.043	%32.17
Estein90(10)	90	6.320064	36	8.285067	30	35.571	%31.09
Estein100(10)	100	6.719510	38	8.803421	35	58.188	%31.01

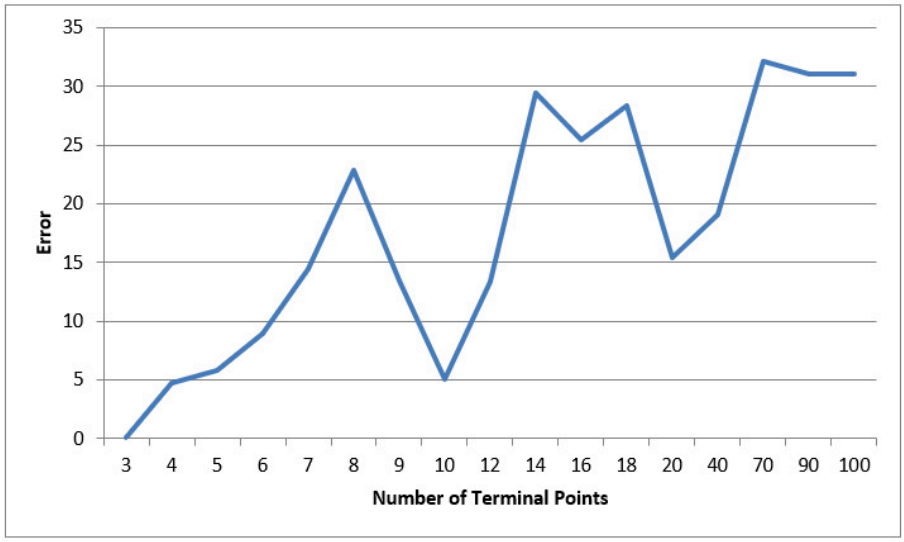

Figure 2: The relation between the number of the terminal points and error for the GKESTP.

Table 3: Results of two methods for the datasets with unknown optimal values.

Problem Name	n **Terminal Point Number**	$\triangle_{k-means}$ **Optimal Value**	**Steiner Point Number with KESTP**	$\triangle_{globalk-means}$ **Resulting Value by k-means Algorithm**	**Steiner Point Number with GKESTP**	**Error (%)**
Estein250(5)	250	14.947258	106	14.554118	69	-%2.63
Estein250(10)	250	14.943349	107	15.781254	57	%5.61
Estein500(5)	500	19.625697	393	20.569492	123	%4.81
Estein500(10)	500	21.054484	160	19.957933	124	-%5.21
Estein1000(5)	1000	33.172557	240	32.106747	188	-%3.31
Estein1000(10)	1000	34.376048	473	33.772009	160	-%1.75

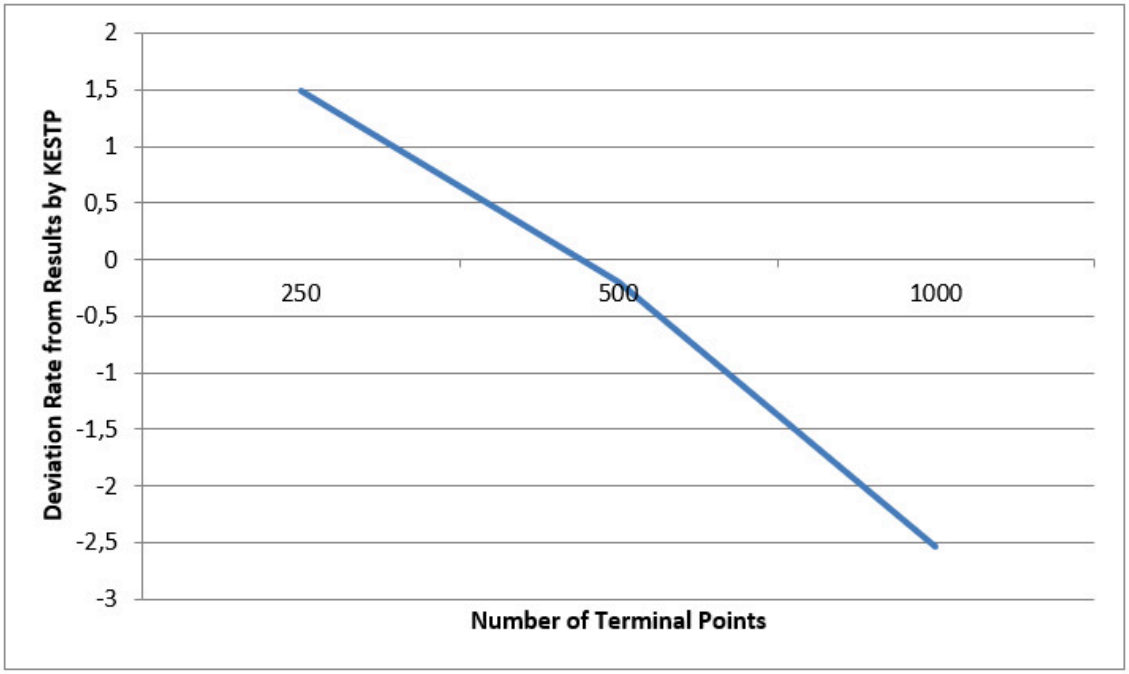

Figure 3: The relation between the results by KESTP and GKESTP for the datasets with unknown optimal values.

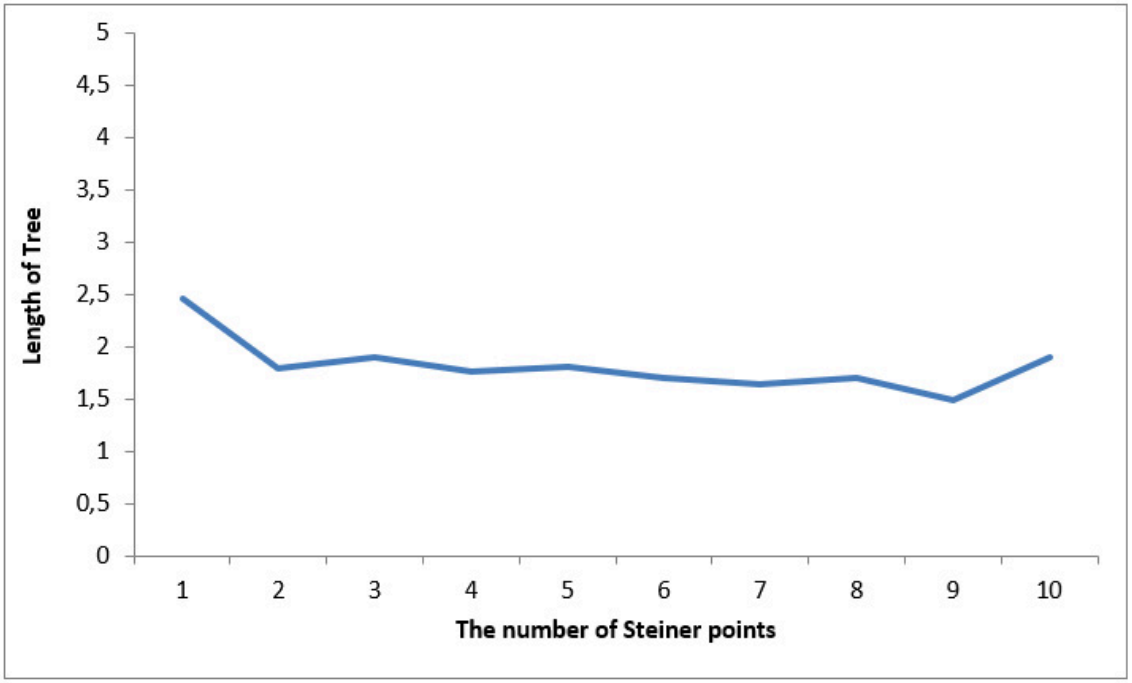

Figure 4: The relation between the length of tree obtained by KESTP and the number of Steiner points for problem Estein1(40).

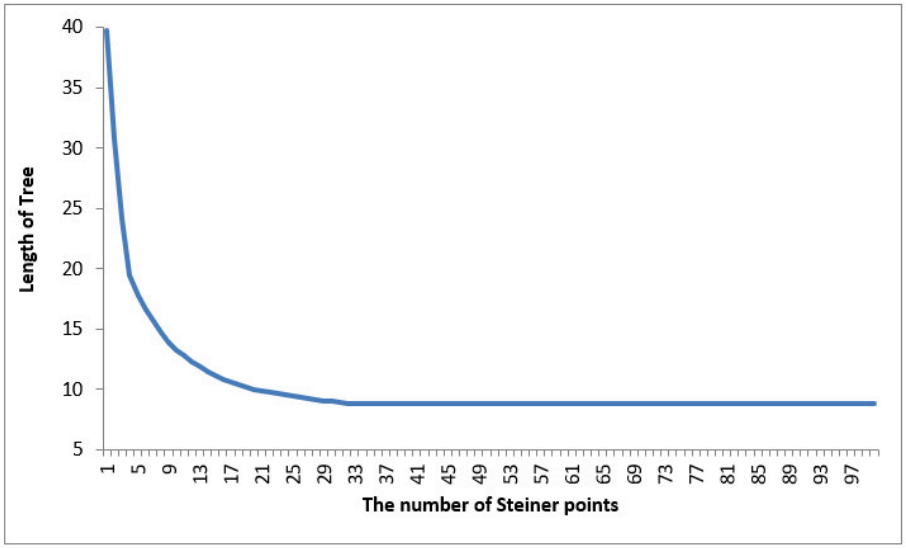

Figure 5: The relation between the length of tree obtained by GKESTP and the number of Steiner points for problem Estein100(10).

these 6 problems. The error in Figure 5 was computed as:

$$error = 100 \times \frac{\triangle_0 - \triangle_{best}}{\triangle_{best}}$$

Here, $\triangle_{best}$ is the obtained result KESTP and $\triangle_0$ is the other obtained result from GKESTP. More specifically, the error rate is the deviation rate from results obtained by KESTP. If the error is negative, this means that the results obtaines with GKESTP are beter than results obtained with KESTP. Obviously, as the number of terminal points increases, GKESPT gives beter results that KESTP. This conclusion is visualized in Figure 6.

The number of Steiner points is given in Figure 1, 3 and 5. While this value were found, Steiner points were found from 1 to n for n terminal points and the number of Steiner points that gives the shortest-length tree is stored. Since there is no formulation for the number of Steiner points, we made clustering for up to the number of terminal points. We visualized this operation for problem Estein1(40) in Figure 7 and for problem Estein100(10) in Figure 8. Comparison in Figure 7 was made for KESTP. The length of tree is minimum for 9 Steiner points, so we consider 9 as the number of Steiner points. Comparison in Figure 5 was made for GKESTP. Obviously, after the number of steiner points is 35, the tree length does not change. More formally, after a certain number, although the number of Steiner points increases, the length of the tree does not improve. Therefore, we consider this number as the number of Steiner points and place this value in the corresponding column in Figure 1, 3 and 5.

6 Conclusions

In this paper, we proposed two novel methods named KESTP and GKESTP for Euclidean Steiner tree problem which is an important combinatorial optimization problem. Firstly, these methods find cluster centers using the k-means and global k-means algorithms. Then, a tree is obtained by using the PRIM's algorithm for these cluster centers. Finally, the minimum length tree among the obtained trees is considered as the solution of the problem.

We have presented the results of numerical experiments on 28 data sets in the literature. Results of numerical experiments show that the proposed algorithms is effective for finding approximate solutions for Euclidean Steiner tree problem. The GKESTP algorithm outperformed

than the KESTP algorithm, especially for large scale problems. However the GKESTP algorithm requires more computational efforts than the KESTP algorithm.

References

Bagirov, A.M., Mardaneh, K. (2006, December). Modified global k-means algorithm for clustering in gene expression data sets. In *Proceedings of the 2006 Workshop on Intelligent Systems for Bioinformatics*, Volume 73 (pp. 23-28).

Beasley, J.E. (2017). OR-Library, December 2017, 1990. Available on `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`.

Beasley, J.E. (1992). A heuristic for Euclidean and rectilinear Steiner problems. *European Journal of Operational Research, 58*(2), 284-292.

Bondy, J.A., Murty, U.S.R. (1976). *Graph Theory with Applications* (Vol. 290). London: Macmillan.

Chlebík, M., Chlebíková, J. (2008). The Steiner tree problem on graphs: Inapproximability results. *Theoretical Computer Science, 406*(3), 207-214.

Dreyer, D.R., Overton, M.L. (1998). Two heuristics for the Euclidean Steiner tree problem. *Journal of Global Optimization, 13*(1), 95-106.

Fampa, M., Anstreicher, K.M. (2008). An improved algorithm for computing Steiner minimal trees in Euclidean d-space. *Discrete Optimization, 5*(2), 530-540.

Fampa, M., Lee, J., & Maculan, N. (2016). An overview of exact algorithms for the Euclidean Steiner tree problem in n-space. *International Transactions in Operational Research, 23*(5), 861-874.

Guichard, D. (2017). *An introduction to combinatorics and graph theory.* Whitman College-Creative Commons.

Gilbert, E.N., Pollak, H.O. (1968). Steiner minimal trees. *SIAM Journal on Applied Mathematics, 16*(1), 1-29.

Hakimi, S.L. (1971). Steiner's problem in graphs and its implications. *Networks, 1*(2), 113-133.

Hanan, M. (1966). On Steiner's problem with rectilinear distance. SIAM *Journal on Applied mathematics, 14*(2), 255-265.

Hartigan, J.A., Wong, M.A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *28*(1), 100-108.

Jones, J., Harris Jr, F.C. (1996, June). A Genetic Algorithm for the Steiner Minimal Tree Problem. In *Proceedings of ISCA's International Conference on Intelligent Systems.*

Karpinski, M., Zelikovsky, A. (1997). New approximation algorithms for the Steiner tree problems. *Journal of Combinatorial Optimization, 1*(1), 47-65.

Koch, T., Martin, A. (1998). Solving Steiner tree problems in graphs to optimality. *Networks: An International Journal, 32*(3), 207-232.

Likas, A., Vlassis, N., & Verbeek, J.J. (2003). The global k-means clustering algorithm. *Pattern recognition, 36*(2), 451-461.

MacQueen, J. (1967). Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability* (pp. 281-297).

Melzak, Z.A. (1961). On the Steiner problem. *Canad. Math. Bull.*, 4, 143-148.

Ordin, B., Bagirov, A.M. (2015). A heuristic algorithm for solving the minimum sum-of-squares clustering problems. *Journal of Global Optimization, 61*(2), 341-361.

Qu, R., Xu, Y., Castro, J.P., & Landa-Silva, D. (2013). Particle swarm optimization for the Steiner tree in graph and delay-constrained multicast routing problems. *Journal of Heuristics, 19*(2), 317-342.

Seda, M. (2001). Computing Near-Optimal Solutions to the Network Steiner Tree Problem Using Approximate and Heuristic Techniques: In *Proceedings of the 13th International Conference Process Control*, Vol. 1.

Smith, W.D. (1992). How to find Steiner minimal trees in euclideand-space. *Algorithmica, 7*(1), 137-177.

Soothill, G. (2010). The euclidean steiner problem. *Report, Department of Mathematical Science, Durham University, England (Undergraduate Departmental Prizes).*

Winter, P., Zachariasen, M. (1996). Large Euclidean Steiner minimum trees in an hour. Datalogisk Institut, Köbenhavns Universitet.

Whittle, D., Brazil, M., Grossman, P.A., Rubinstein, J.H., & Thomas, D.A. (2022). Solving the prize-collecting Euclidean Steiner tree problem. *International Transactions in Operational Research, 29*(3), 1479-1501.

Xu, R., Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks, 16*(3), 645-678.